

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,300

Open access books available

130,000

International authors and editors

155M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Management of Software-Defined Networking Powered by Artificial Intelligence

Jehad Ali and Byeong-hee Roh

Abstract

Separating data and control planes by Software-Defined Networking (SDN) not only handles networks centrally and smartly. However, through implementing innovative protocols by centralized controllers, it also contributes flexibility to computer networks. The Internet-of-Things (IoT) and the implementation of 5G have increased the number of heterogeneous connected devices, creating a huge amount of data. Hence, the incorporation of Artificial Intelligence (AI) and Machine Learning is significant. Thanks to SDN controllers, which are programmable and versatile enough to incorporate machine learning algorithms to handle the underlying networks while keeping the network abstracted from controller applications. In this chapter, a software-defined networking management system powered by AI (SDNMS-PAI) is proposed for end-to-end (E2E) heterogeneous networks. By applying artificial intelligence to the controller, we will demonstrate this regarding E2E resource management. SDNMS-PAI provides an architecture with a global view of the underlying network and manages the E2E heterogeneous networks with AI learning.

Keywords: Software-defined networking, Machine learning, 5G, Networks management, Artificial intelligence

1. Introduction

Due to the rapid development of Internet technology, network terminals have been widely spread. However, traditional network architectures have failed to adapt to future advances in communication and Internet technologies, resulting in heterogeneous networks. As a result, the existing network infrastructure was unable to keep up with the rapid changes of the Internet. A key feature of traditional network architectures is that the data and control planes are tightly coupled, which has some limitations. For example, if you want to change the network configuration, you need to configure each device independently across the entire network which is a daunting task.

Similarly, vendors are reluctant to provide the internal details of the device to developers and users, as changes in the configuration of existing networking devices can lead to malfunctions in the network. In addition, the protocol is strongly built into the firmware of network devices. These limitations hinder

network innovation due to proprietary hardware and lack of testing for innovative networking solutions due to their distributed nature. It also increases the management workload and the overall cost of network management.

On the other hand, Software Defined Networking (SDN) [1–5] has revolutionized network management by separating data and control planes. The data plane is composed of forwarding devices, for example routers, switches, etc. Its main functions are forwarding the packets according to the policies of the controller. If the destination of the arrived packets is not found in the forwarding devices, then those packets are sent to the controller by the data plane. The control plane, however, is implemented through intelligent SDN controllers such as OpenDaylight (ODL), Open Networking Operating System (ONOS), POX and RYU [6]. Control plane obtains the status of the underlying network and defines the policies for the packets arriving on the forwarding devices. It then pushes the updated rules to the data plane. The separation of data and control planes has shifted network complexity from networking devices to smart SDN controllers. Thus, the network can be programmed through the application running on the controller and the underlying network is abstracted from the applications [7]. The innovative concept presented by SDN has the great advantage of flexible and efficient network configuration, network management and operation. Therefore, SDN is expected to be an excellent choice for the next generation of telecommunication networks and Internet technologies. Because of these benefits, large information technology organizations such as Facebook, Amazon, and Google have implemented SDN to connect remote data centers [8, 9].

The internet has grown in recent years. As a result, there is a huge increase in the amount of network traffic. Because the accuracy of machine learning algorithms depends mainly on the availability of historical data. There is therefore an increasing tendency towards the use of machine learning techniques. Because the accuracy of machine learning algorithms increases with sufficient data. For this reason, researchers now prefer to apply machine learning solutions because, once trained on the available data, the trained model generates accurate results on the new data through learning experience. The introduction of 5G heterogeneous networks and the rapid ubiquitous use and growth of Internet data processing requirements are rapidly increasing as a result of a dramatic increase in the number of connected devices. For example, the heterogeneous IoT devices in 5G runs different protocols and various technologies results in increasing the traffic load [10]. In addition, there is a need for self-organization and demand-based networks to deal with huge amounts of data. SDN was therefore at the heart of the growing needs of such applications due to their programming, orchestration, and automation characteristics [11].

The SDN has been successfully deployed in data centers and enterprise traffic engineering networks across remote data centers. However, the adoption of SDN in the modern and global Internet still presents a number of challenges that need further investigation. As the internet is scaling and the traffic on the underlying network is dynamically changing. The application of an optimum policy for the underlying network should therefore be adapted in line with the radical changes in the internet. One of the problems in SDN is the configuration of the control plane, because the manual configuration is a costly task, because the traditional SDN approach [12–15] is not optimal in selecting the optimum policy for the underlying network. In addition, repeatedly reconfiguring the policy according to changes in the network will require the control plane to be reconfigured. One of the main issues, therefore, is the automatic orchestration of the control plane [16]. Because the rigid configuration of the control plane will have problems in the optimal configuration of the policy.

Another issue is the end-to-end (E2E) quality-of-service (QoS) performance of heterogeneous network providers. If the same provider manages SDN controllers, user applications and forwarding devices on the enterprise network, then, the network status of the underlying devices is readily available for upper-layer applications. However, the Internet consists of different providers where end-users, applications and service providers are often heterogeneous. As a result, the status of the network is not directly available for applications running on the upper layers.

Several solutions have been proposed to address the issue of the allocation of E2E resources [17–23]. However, they depend on the traditional and manual configuration of the control plane. i.e., once a policy has been defined for the underlying network. The behavior of the network is then controlled accordingly, regardless of the scale of the network or the dynamic changes. The policy of controlling the network is therefore not always optimal. Moreover, these solutions do not provide effective management of the SDN due to scaling up, increasing network complexity and dynamic changes. There is therefore a need to find a global optimal solution with an excellent value for the objective functions. We therefore propose a software-defined networking management system powered by AI (SDNMS-PAI) architecture to auto-configure policy management and E2E resource allocation.

The advantage of AI based architecture is that the AI agent will interact with the underlying network through the SDN controller for pushing the global optimal policy flow rules in the forwarding devices. The controller will share the network status information with the AI agent and based on real time status of the network the AI agent will find the most appropriate actions to be taken. The actions will be pushed as the flow rules in the forwarding devices. AI can be used to bring a closed-loop control of the SDN. The closed-loop control incorporates collection of data, analytics, and subsequent actions that are all based on the results of the analytics [24]. All components of the closed loop can be improved and enhanced by means of AI to improve the speed, accuracy and, ultimately, the effectiveness of the closed loop control.

The main contributions of this chapter are summarized as follows:

- We leverage the hierarchical SDN architecture to provision the E2E QoS for heterogeneous networks and build a centralized intelligent agent with global E2E view aiming at learning the global optimum policy through interaction with the data plane.
- We apply Q-learning where the learning agent obtains the states of the underlying network and provisions the E2E resource allocations for a service request in the heterogeneous network domains with several QoS classes on the E2E path.
- We demonstrate the proposed SDNMS-PAI with a use case for E2E resource allocation i.e. E2E QoS provisioning.
- Moreover, we evaluate the E2E delay, jitter, packet loss ratio (PLR), and E2E degree of correspondence (DC) [25] ratio for service requests in a hierarchical SDN architecture with an AI agent.

2. AI powered SDN architecture

In this section, an overview of the proposed SDNMS-PAI is provided. First, we introduce the three planes of the SDN architecture and explain them with a pictorial

diagram. Then, we introduce the hierarchical SDN architecture for the allocation of E2E resources and the deployment of AI enabled learning. The hierarchical control plane consists of two levels of hierarchy of local and global controllers. Then we develop the SDNMS-PAI architecture for the E2E view and the resource allocation leveraging Q-learning. The proposed architecture consists of a hierarchical control plane with a global E2E view and leverages Q-learning to manage E2E resources in SDN in a smart way.

2.1 Hierarchical control plane SDN architecture powered by AI

In this subsection, we first introduce an SDN and a hierarchical architecture followed by an AI powered SDN architecture. The SDN consists of data, control, and application planes. **Figure 1** [26] shows the typical SDN architecture. Forwarding devices like routers and switches are part of the data plane. The centralized controller is part of the control plane. At the top is the application plane where different applications can be deployed and executed for a variety of purposes, such as routing, load balancing, security, and monitoring. The controller shall act as a strategic control point for the underlying network. However, several issues arise from a single controller in the SDN. For example, if the controller fails due to a software or hardware problem, the entire network that depends on the controller will collapse.

In addition, the controller will experience a performance bottleneck if the number of switches in its domain increases or the request messages towards it increases. Furthermore, traffic loads are not evenly distributed over the network. As a result,

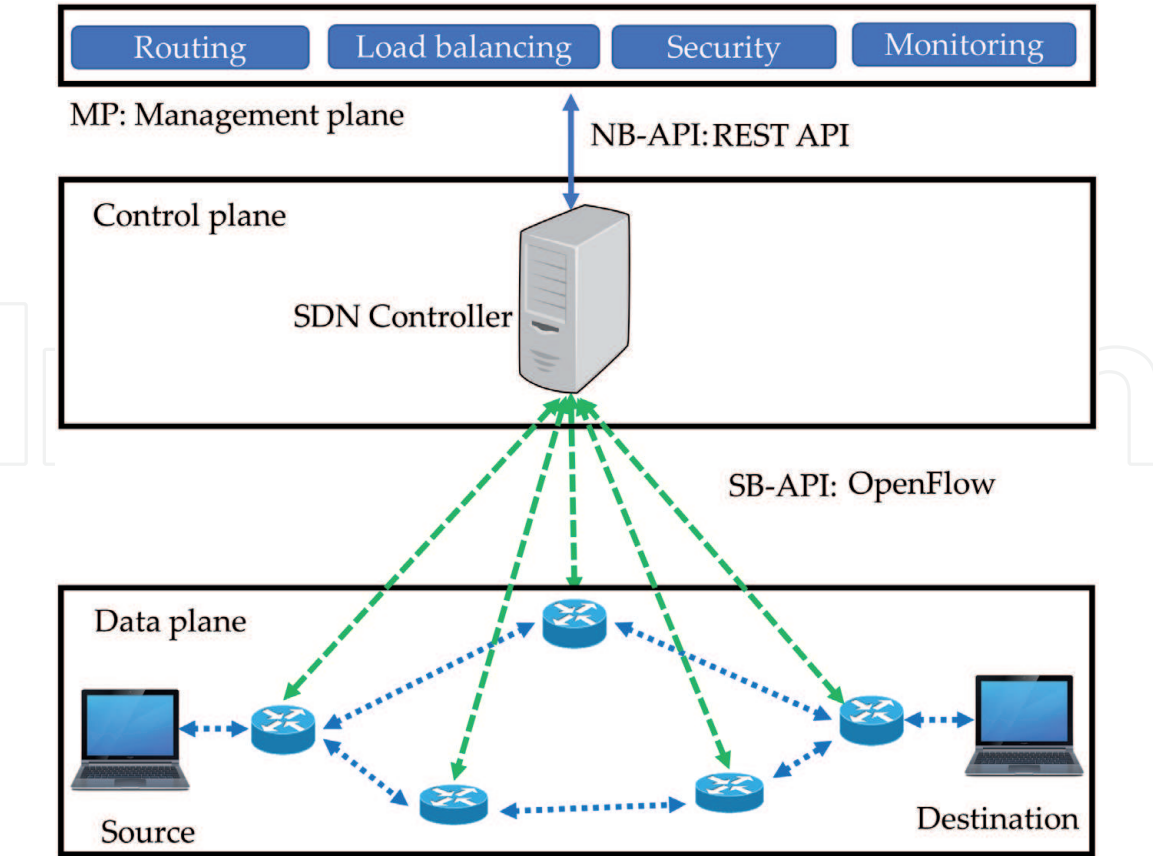


Figure 1.
SDN architecture [26].

multiple controllers should be used for viewing the E2E network. However, if there are multiple heterogeneous domains, there is a need for consistency and collaboration between domains for the provisioning of E2E QoS.

Figure 2 shows the hierarchical control plane SDN architecture. In the proposed architecture there are local controllers which has access of the data planes of the local domains. Global controllers (GCs) in the hierarchical control plane architecture have access to the global view of physically distributed local data plane switches. The hierarchical architecture of SDN controllers integrates autonomous domains with hierarchical associations. Multiple domains are integrated with the hierarchical architecture of the controller, where the local domain controllers (LCs) coordinate via the GC. By applying hierarchical architecture, new services can be easily managed and deployed in domains that coexist on the E2E path between the source and the destination [27] nodes.

The tasks handled by the controller are propagated from the lower LC layer to the upper GC layer, which reduces computational complexity. The hierarchical control plane with a global view reduces the E2E delay as the network scales [28]. In the proposed architecture, the GC acts proactively to set up the E2E path and therefore reduces the delay in setting up the path (the delay in setting up the path and pushing the flow entries into the switches) [29]. The hierarchical architecture enables communication between multiple LCs with a variety of equipment. The effectiveness of the hierarchical control plane for effective collaboration between heterogeneous tactical networks with a guaranteed QoS has been demonstrated in [30, 31]. The rewards for state action pairs in the Q-learning are therefore more accurate than the local view states because these rewards with a hierarchical architecture reflect the E2E view of the underlying network.

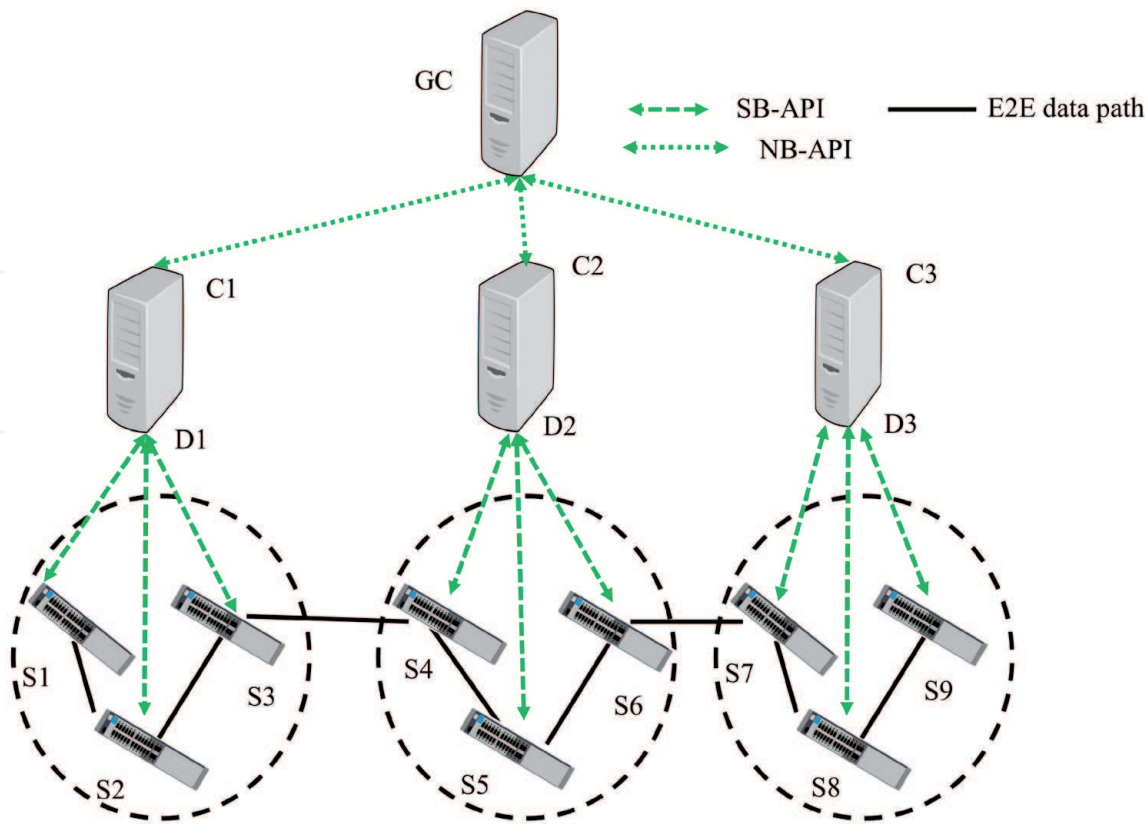


Figure 2.
SDN architecture with a global view of the E2E network [32, 33].

In our proposed SDNMS-PAI, a hierarchical control plane architecture is employed to construct a completely global view and control for geographical distributed network and build a global AI agent through the global control plane to generate a network control policy via reinforcement learning algorithms. The SDNMS-PAI can intelligently control and optimize a network to meet the differentiated network requirements in a large-scale dynamic network. In the following subsections, we describe the proposed AI enabled SDN architecture from bottom to top. The SDNMS-PAI is shown in **Figure 3**.

2.1.1 Data plane

Data plane in the SDNMS-PAI consists of the forwarding devices (known as the infrastructure or the underlying network). The matching of the packets in the data plane and the actions take place according to the forwarding rules that are defined in a flow Table. A flow table comprises of several flow entries. The packet header information is matched with the flow entries in the flow table. Each flow entry has three mandatory fields, i.e., header, action, and counter. **Table 1** is an example of a flow table in which the first row contains header fields and second and onward rows contain flow entries.

When a new packet arrives on the ingress port of a switch, the matching process starts, if a packet has a destination IP address starting with 172.10.X.X then forward it to port number 8 and counter 201 will be updated. Similarly, the third row (with source IP address: 10.10.1.X) explains if a packet has the same source and destination port number (X) then drop it. If the rules for the new packet do not exist in the flow table then the switch sends a Packet_In message to the controller and the destination will be returned by the controller to the forwarding device (Packet_Out message) and the flow rules will be updated in the flow table, respectively. In contrast to traditional networks where the decision about the routing takes place in the tightly couple distributed networking devices. Herein, in the SDNMS-PAI, the information of the network is collected via the LCs which is used by the AI enabled

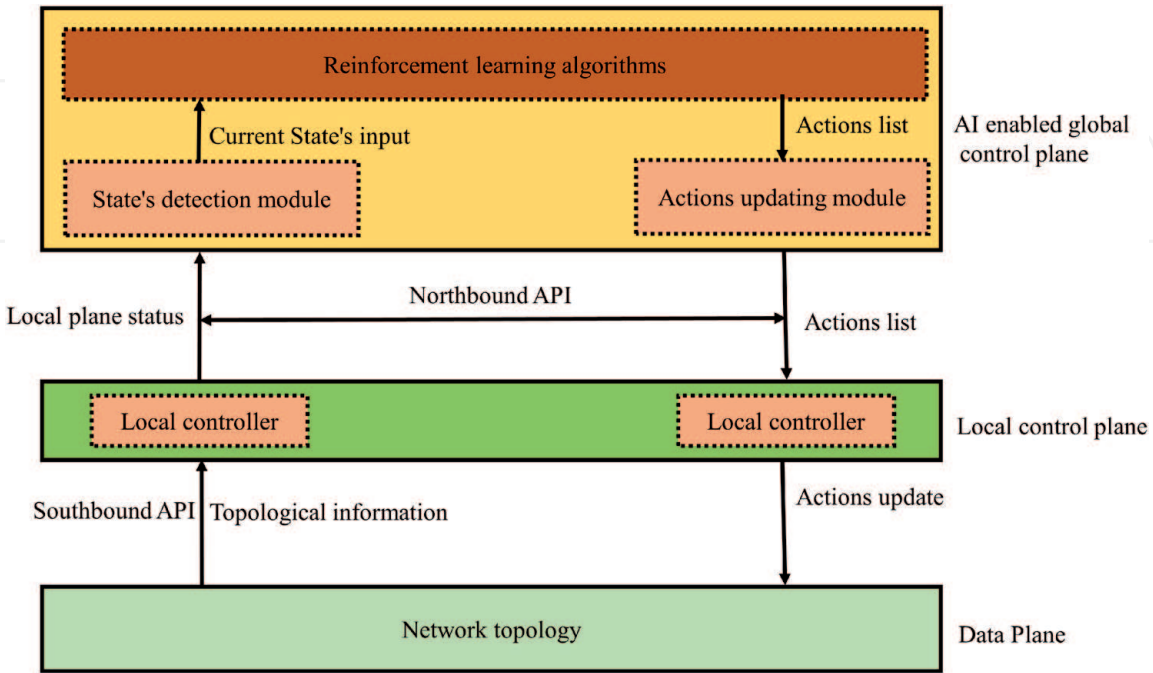


Figure 3.
AI powered SDN architecture for E2E resource allocation.

Source (IP address)	Destination (IP address)	Source (Port)	Destination (Port)	Action	Counter
X	172.10.X.X	X	X	Port 8	201
X	X	10	40	Drop	80
10.10.1.X	X	X	X	Drop	90
X	X	30	70	Port 3	100

Table 1.
An example of the flow table entries.

global control plane (AIGCP) for deciding about the global optimum policy and E2E resources allocation.

2.1.2 SB-API

The Southbound Application Programming Interface (SB-API) provides an interface for data interaction with the local control plane. There are several protocols available for the interaction of the two planes, but the most popular is OpenFlow. OpenFlow provides a secure interface for communication between the controller and the switch. The status of the network topology and the policies for action from the global control plane are communicated to the data plane via the SB-API in the SDNMS-PAI. The White Paper [34] describes the advantages and flexibility of OpenFlow for the programming of forwarding devices. The concept of OpenFlow originated from Stanford University, and the OpenFlow Networking Foundation (ONF) consortium now performs the standardization tasks of OpenFlow.

2.1.3 Local control plane

The data plane switches of each domain are connected to the LCs on the E2E path. The LCs interact with the data plane through SB-APIs. The AIGCP dynamically obtains the underlying network status from the LCs; therefore, it has access to the global topology. As a result, the AIGCP will provide resources from local controllers upon the arrival of a service request. LCs work together through GC, and service level agreements (SLAs) are exchanged through it. Each LC is equipped with a traffic flow template (TFT) module [35] containing the source and destination port numbers, the Internet Protocol (IP) addresses and the QoS parameters. The data collected will be used by the AIGCP for the allocation of E2E resources.

2.1.4 NB-API

The northbound application programming interface (NB-API) functions as a communication interface between the local control and AIGCP. The local control plane functions as a bridge between the forwarding devices and AIGCP utilizing the representational state transfer (REST) API. Similarly, the operational statistics (e.g., about the flow entries) from the data plane are available via this API to the global control plane AI agent. Reinforcement learning algorithms running in the global control plane communicates with the local control plane through this API and the corresponding actions are delegated to the data plane. These actions represent the behavior of the reinforcement learning algorithms executed in the global control plane. For example, a firewall application implements policies for

controlling the ingress and egress packets passing through the network. Therefore, the data plane devices will forward or block the traffic according to the rules defined in the application. Similarly, a load balancing algorithm will control the traffic through monitoring congestion in different paths of the network. Herein, we employ the Q-learning for E2E QoS provisioning.

2.1.5 AI enabled global control plane

The purpose of the AIGCP is to generate global optimum policies leveraging the global view from the hierarchical SDN architecture. In the SDNMS-PAI paradigm, the AIGCP leverage of hierarchical SDN architecture to obtain the global view as well as control of the E2E network. The state detection module in the global control plane has the global view of the E2E network status which helps the AI agent to make decisions about the global optimum policy based on the E2E view. It feeds the AI agent with the information about the states of the E2E network.

2.1.6 Optimal policy learning mechanism

The local controllers obtain the QoS information (such as the delay, jitter, and PLR) from the data plane devices for all the service requests and the service classes on the E2E paths. The service requests and service classes are shown in **Table 2** [36] and **Table 3** [37]. The service request is a combination of the E2E delay, jitter, and PLR for an application. An example of the offered service classes in 5 E2E domains is shown in **Table 3**. Each local controller shares this information with the global controller. Thus, global controller has the E2E view of the network.

Reinforcement learning with Q-learning enabled AI agent is used to maximize the rewards for an agent. Q-learning is one of the methodologies to leverage reinforcement learning. It does not require a model of the environment, and it can cope with problems utilizing stochastic transitions with rewards, without demanding adaptations. For a finite Markov decision process (FMDP), Q-learning computes an optimal policy aiming to maximize the expected value of the accumulated reward over every as well as all successive steps, beginning from current state. Q-learning can find an optimal action-selection policy for any given FMDP, given infinite exploration time along with partly-random policy [38]. Q is the function name that the algorithm learns with the maximum expected rewards for an action taken in a given state [39].

If the service request meets the end-to-end QoS demand for a state action pair, a high reward factor is assigned. For this purpose, the DC ratio is checked for the state action pair. The DC ratio denotes whether the QoS requirements are meeting for a service request or not. For example, if the application service request E2E demand for delay is 150 and the service classes offer a delay of 40, 20, 15, 0 and 45 on the E2E path, then the ratio will be 150/120 i.e., 1.25. Hence, if the $DC > 1$ it is awarded

Metric	Service Requests for an application		
	1	2	3
Delay (ms)	150	200	400
Jitter (ms)	60	60	80
PLR	10^{-4}	10^{-3}	10^{-3}

Table 2.
An example of the E2E service requests.

Domain	QoS Class	Offered Delay (ms)	Offered jitter (ms)	Offered PLR
1	1	40	10	10^{-5}
	2	80	30	10^{-4}
	3	120	0	10^{-4}
2	1	20	15	10^{-6}
	2	50	20	10^{-5}
	3	70	30	5×10^{-5}
	4	120	0	10^{-4}
3	1	15	10	10^{-6}
	2	50	30	10^{-5}
4	1	12	6	10^{-5}
	2	0	0	10^{-4}
5	1	45	5	10^{-5}
	2	100	15	10^{-4}
	3	120	40	10^{-4}

Table 3.
An example of the service classes on the E2E path passing through five domains.

a high Q value for the service request. On the contrary if the $DC < 1$, the reward is low for the state action pair for that service request. This process continues until all the possible source to destination paths are explored and checked for the DC value against each state action pair.

3. Use case

Herein, we describe a scenario in which we can employ our proposed SDNMS-PAI for modeling the behavior of the network. We provide an example in the context of QoS service classes allocation, where the SDNMS-PAI is used to make smart choices in order to choose the best service classes on the E2E routing path to meet the E2E QoS requirements. Moreover, based on the Q-learning rewards more excellent service classes are selected in future. The traditional design of the internet mainly focusses on the reliability of services [16]. However, with 5G and beyond networks the requirements for applications have changed, and the applications demands for low latency with high data rates. Further, it is imperative whether the E2E QoS is according to the application service requests. Moreover, with heterogeneous networks on the path from source to destination, there exists several service classes in each domain. Hence, meeting the E2E QoS requirements for the applications service requests is a challenging problem.

Service class mapping mainly involves service classes allocation on the E2E path that meets the QoS demands of different service requests. The typical E2E service classes request for each application are different as shown in **Table 2**. For example, for application 1 the service requests are different than from application 2 and so on. Several solutions [40–42] have been proposed by researchers for service class mapping to meet the E2E QoS requirements for the applications. Furthermore, the mapping of the service classes is a challenging task with respect to meeting the E2E service needs due to the local view of the network state information in the domains.

4. Results and discussion

Results of the proposed SDNMS-PAI are compared with existing ones i.e., software-defined networking with no artificial intelligence (SDN-NAI) [32]. There are 5 domains on the E2E path and two layers of the controllers i.e., local controllers and a global controller. We consider delay, jitter, and PLR as the primary QoS parameters in every domain. Controllers of the five domains are assigned to 50 nodes according to the controller placement in [43].

Figure 4 compares the E2E delay (in milliseconds (ms)) from source to destination for the SDN-NAI i.e., SDN with no artificial intelligence enabled global control plane and our proposed SDNMS-PAI with. We can see that the delay for the initial service requests is greater for the SDNMS-PAI because the AI agent explores the E2E paths from source to destination for the optimal service classes. However, as the AI agent learns about the global optimum policy, then the delay decrease as compared to SDN-NAI which is shown in the 3rd, 4th and 5th domains. Initially the service request rates are smaller hence the delay is low however with increasing the service request rate the delay increases because of the consumption of the available bandwidth resources on the E2E paths.

The results in **Figure 5** show that E2E jitter (ms) from source to destination for an SDN-NAI compared with SDNMS-PAI. The figure reveals that the jitter for the initial service requests is greater for the SDNMS-PAI due to the AI agent exploring the E2E paths from source to destination to find the optimal service classes. However, as the AI agent becomes more proficient in learning about the global optimum policy, then the jitter decreases as compared to SDN-NAI, which is shown in the 3rd, 4th, and 5th domains. Initially, with lower service request rate the jitter is low since each service request requires only a portion of the available bandwidth on an E2E path. With increasing the service request rate, however, the jitter will increase because of the bandwidth resources used in each service request.

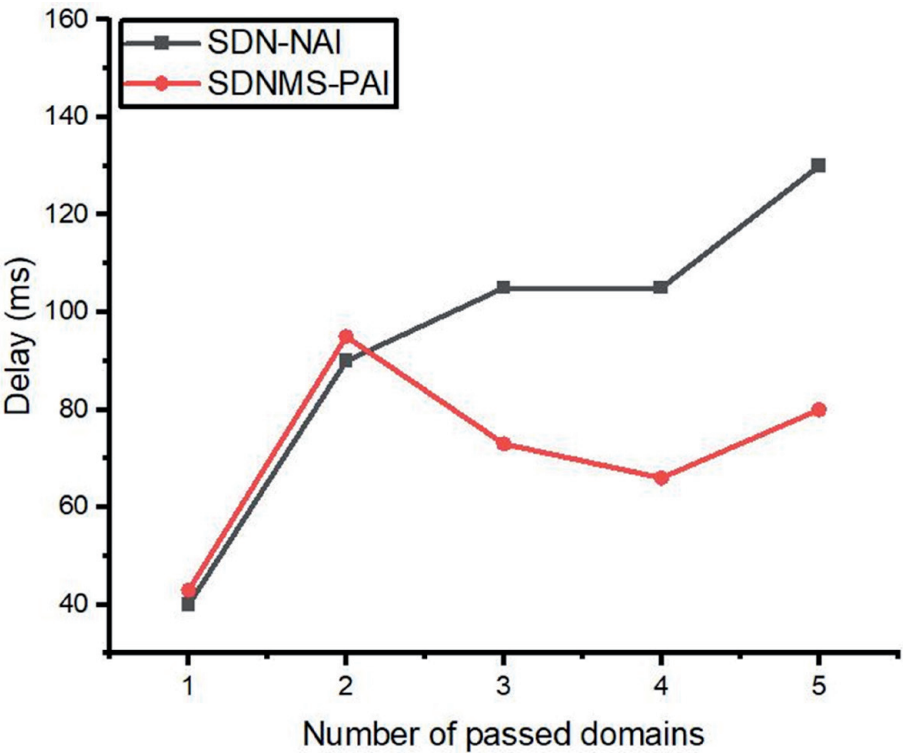


Figure 4.
E2E delay from source to destination with increasing service requests passing through five domains.

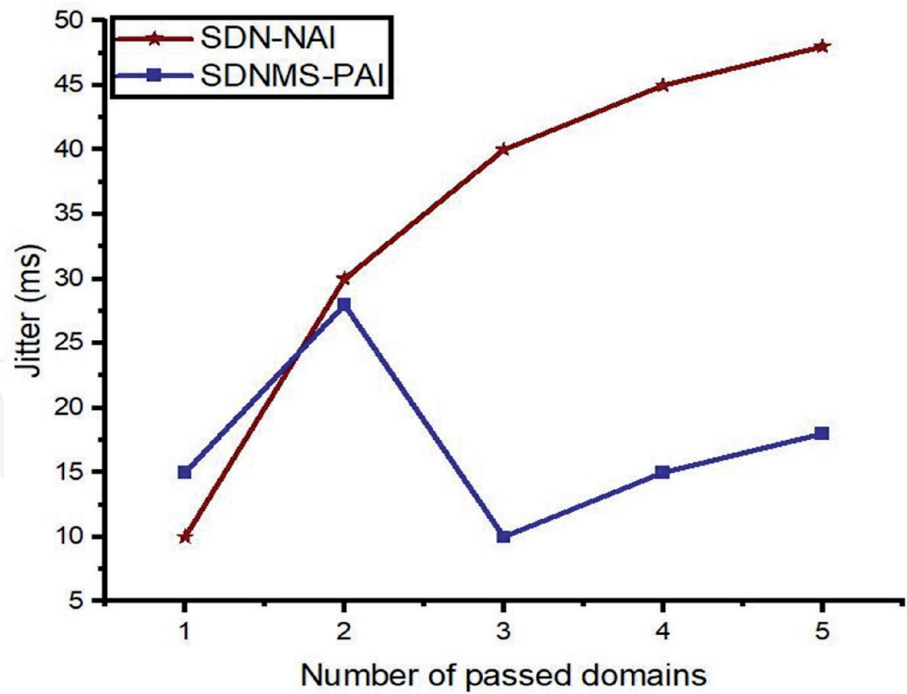


Figure 5.
E2E jitter from source to destination with increasing service requests passing through five domains.

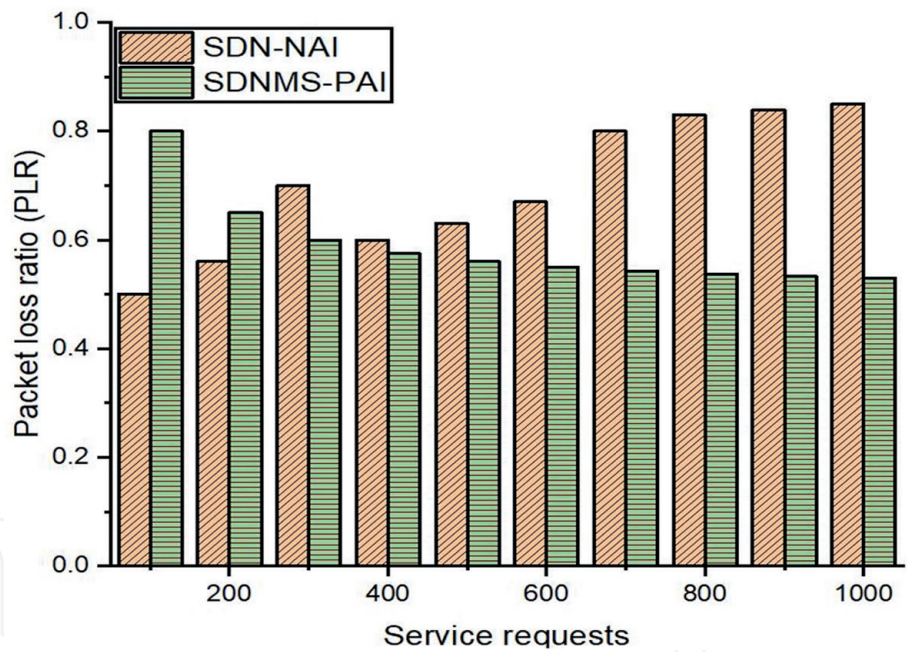


Figure 6.
Packet loss ratio (PLR) with increasing service request rate.

Figure 6 compares the PLR with increasing the service request rate. Herein, the PLR is the ratio of the number of received packets divided by the total number of packets against each service request from source to destination. We can see from **Figure 6** that the PLR is initially high for the SDNMS-PAI however as the AI agent obtains a global optimum then the PLR does not increase in the same rate with SDN-NAI. However, the overall PLR increase with increasing the service request rate because the available resources in the network gets occupied.

Figure 7 shows a comparison of the E2E DC ratio for SDNMS-PAI and SDN-NAI. We can see from the figure that the SDN-NAI DC ratio was initially higher than the SDNMS-PAI. However, as the AI agent learns, the DC ratio for the proposed scheme is much higher than the SDN-NAI ratio. The basic reason is that, as the service

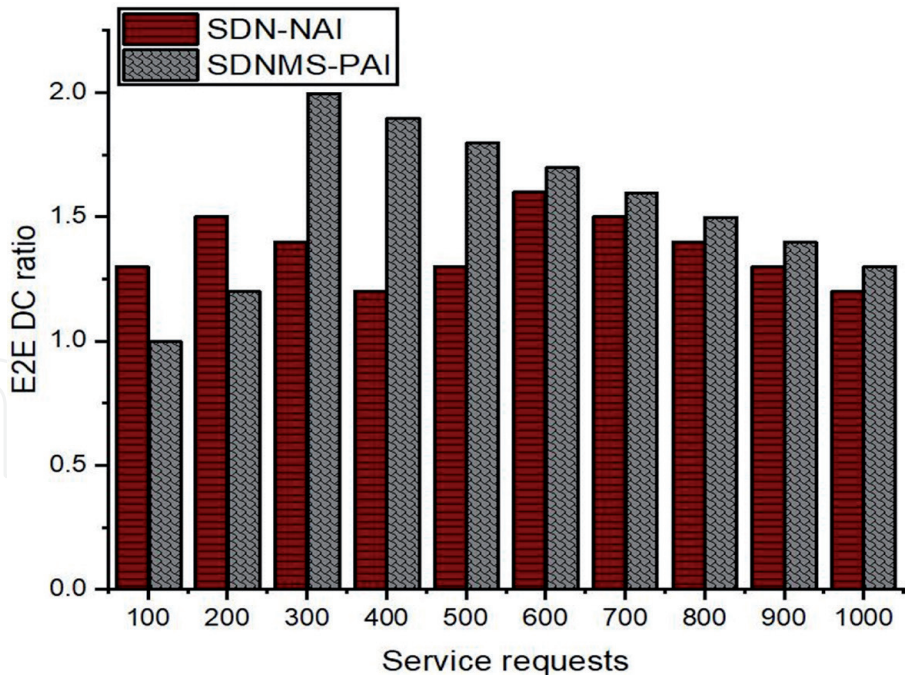


Figure 7.
E2E DC ratio against service requests.

requests increase, the overall DC ratio becomes low due to the consumption of the available bandwidth on the E2E pathways. Nevertheless, the E2E DC ratio is still 1 or greater than 1 for the proposed SDNMS-PAI, which means that it satisfies the QoS requirements for the application service request. In addition, it overcomes the SDN-NAI in E2E DC ratio.

5. Conclusions

In this chapter we proposed SDNMS-PAI for the E2E resource allocation i.e., service classes allocation for the E2E service requests. As the distributed management and tight coupling of control and data planes limit the control and global view of network resources. Moreover, the E2E resources in heterogeneous networks cannot be provisioned. Hence, in this chapter we proposed the hierarchical SDN architecture because a single controller with manual configuration of the control plane led to failure and restricts the optimal policy. Moreover, we provided a use case example with service requests and service classes. Furthermore, the SDNMS-PAI scheme employed in a hierarchical SDN architecture with AI agent in the global control plane overcomes the SDN-NAI in terms of E2E delay, jitter, PLR, and DC ratio.

Acknowledgements

This was supported by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program, supervised by the Institute for Information and Communications Technology Planning and Evaluation, under Grant IITP-2021-2018-0-01431.

Conflict of interest

The authors declare no conflict of interest.

IntechOpen

IntechOpen

Author details

Jehad Ali and Byeong-hee Roh*
Department of Computer Engineering, and Department of AI Convergence
Network, Ajou University, Suwon, South Korea

*Address all correspondence to: bhroh@ajou.ac.kr

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Ahmad S, Mir AH. Scalability, Consistency, Reliability and Security in SDN Controllers: A Survey of Diverse SDN Controllers. *Journal of Network and Systems Management*. 2021 Jan;29(1):1-59. DOI: 10.1007/s10922-020-09575-4
- [2] Sarmiento D, Lebre A, Nussbaum L, Chari A. Decentralized SDN Control Plane for a Distributed Cloud-Edge Infrastructure: A Survey. *IEEE Communications Surveys & Tutorials*. 2021;:1-1. DOI: 10.1109/COMST.2021.3050297.
- [3] Singh S, Jha RK. A survey on software defined networking: Architecture for next generation network. *Journal of Network and Systems Management*. 2017 Apr 1;25(2):321-74.
- [4] Tadros CN, Rizk MR, Mokhtar BM. Software defined network-based management for enhanced 5G network services. *IEEE Access*. 2020 Mar 12;8:53997-4008.
- [5] Long Q, Chen Y, Zhang H, Lei X. Software Defined 5G and 6G Networks: a Survey. *Mobile Networks and Applications*. 2019;.
- [6] Ali J, Lee S, Roh BH. Performance analysis of POX and Ryu with different SDN topologies. In *Proceedings of the 2018 International Conference on Information Science and System 2018* Apr 27 (pp. 244-249).
- [7] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*. 2008 Mar 31;38(2):69-74.
- [8] Jain S, Kumar A, Mandal S, Ong J, Poutievski L, Singh A, Venkata S, Wanderer J, Zhou J, Zhu M, Zolla J. B4: Experience with a globally-deployed software defined WAN. *ACM SIGCOMM Computer Communication Review*. 2013 Aug 27;43(4):3-14.
- [9] Ali J, Roh BH, Lee S. QoS improvement with an optimum controller selection for software-defined networks. *Plos one*. 2019 May 31;14(5):e0217631.
- [10] Kazmi SA, Khan LU, Tran NH, Hong CS. *Network slicing for 5G and beyond networks*. Springer International Publishing; 2019 May 14.
- [11] Ali J, Roh BH, Lee B, Oh J, Adil M. A Machine Learning Framework for Prevention of Software-Defined Networking controller from DDoS Attacks and dimensionality reduction of big data. *International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea (South), 2020, 515-519, DOI: 10.1109/ICTC49870.2020.9289504.
- [12] Duan Q. End-to-end service delivery with QoS guarantee in software defined networks. *Transactions on Networks and Communications*. 2018; 6(2), p. 10, 2018.
- [13] Mendiola A, Astorga J, Jacob E, Stamos K. Enhancing network resources utilization and resiliency in multi-domain bandwidth on demand service provisioning using SDN. *Telecommunication Systems*. 2019 Jul;71(3):505-15.
- [14] Francesco L, Marchetto G, Risso F, Santuari M, Gerola M. A Proposal for End-to-End QoS Provisioning in Software-Defined Networks. *International Journal of Electrical and Computer Engineering (IJECE)*. 2017; 7(4), 2261-2277. DOI: 10.11591/ijece.v7i4.pp2261-2277

- [15] Egilmez HE, Dane ST, Bagci KT, Tekalp AM. OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks. In Proceedings of the 2012 Asia Pacific signal and information processing association annual summit and conference 2012 Dec 3 (pp. 1-8). IEEE.
- [16] Yao H, Jiang C, Qian Y. Developing Networks Using Artificial Intelligence. Springer International Publishing; 2019 Apr 26.
- [17] Alshaer H, Haas H. Software-Defined Networking-Enabled Heterogeneous Wireless Networks and Applications Convergence. IEEE Access. 2020 Apr 6;8:66672-92.
- [18] Ibarra-Lancheros KS, Puerto-Leguizamón G, Suárez-Fajardo C. Quality of service evaluation based on network slicing for software-defined 5G systems. TecnoLogicas. 2018 Dec;21(43):27-41.
- [19] Bagci KT, Tekalp AM. SDN-enabled distributed open exchange: Dynamic QoS-path optimization in multi-operator services. Computer Networks. 2019 Oct 24;162:106845.
- [20] Das D, Bapat J, Das D. A dynamic QoS negotiation mechanism between wired and wireless SDN domains. IEEE Transactions on Network and Service Management. 2017 Sep 25;14(4):1076-85.
- [21] Joshi KD, Kataoka K. PRIME-Q: Privacy aware End-to-end QoS framework in multi-domain SDN. In 2019 IEEE Conference on Network Softwarization (NetSoft) 2019 Jun 24 (pp. 169-177). IEEE.
- [22] F. Lucrezia, G. Marchetto, F. Risso, M. Santuari, and M. Gerola, "A proposal for End-to-end QoS provisioning in software-defined networks," Int. J. Electr. Comput. Eng. (IJECE), vol. 7, no. 4, pp. 2261-2277, 2017.
- [23] Karakus M, Durresi A. A scalable inter-as qos routing architecture in software defined network (sdn). In 2015 IEEE 29th International Conference on Advanced Information Networking and Applications 2015 Mar 24 (pp. 148-154). IEEE.
- [24] Gilbert M. Artificial Intelligence for Autonomous Networks; 2020 June 30.
- [25] Stojanovic MD, Rakas SV, Acimovic-Raspopovic VS. End-to-end quality of service specification and mapping: The third party approach. Computer Communications. 2010 Jul 1;33(11):1354-68.
- [26] Ali J, Lee GM, Roh BH, Ryu DK, Park G. Software-Defined Networking Approaches for Link Failure Recovery: A Survey. Sustainability. 2020 Jan;12(10):4255.
- [27] Li LE, Mao ZM, Rexford J. Toward software-defined cellular networks. In 2012 European workshop on software defined networking 2012 Oct 25 (pp. 7-12). IEEE.
- [28] Elgendi I, Munasinghe KS, Jamalipour A. A three-tier SDN architecture for DenseNets. In 2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS) 2015 Dec 14 (pp. 1-7). IEEE.
- [29] Khalili R, Despotovic Z, Hecker A. Flow setup latency in SDN networks. IEEE Journal on Selected Areas in Communications. 2018 Sep 19;36(12):2631-9.
- [30] Elgendi I, Munasinghe KS, Mcgrath B. A heterogeneous software defined networking architecture for the tactical edge. In 2016 Military Communications and Information Systems Conference (MilCIS) 2016 Nov 8 (pp. 1-7). IEEE.

- [31] Ali J, Roh BH. Quality of Service Improvement with Optimal Software-Defined Networking Controller and Control Plane Clustering, *cmc-computers materials & continua*. 2021. 67(1), 849-875, DOI:10.32604/cmc.2021.014576
- [32] Ali J, Roh BH. An Effective Hierarchical Control Plane for Software-Defined Networks Leveraging TOPSIS for End-to-End QoS Class-Mapping. *IEEE Access*. 2020 May 11;8:88990-9006.
- [33] Ali J, Roh BH. A Framework for QoS-aware Class Mapping in Multi-domain SDN. In 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) 2019 Oct 17 (pp. 0602-0606). IEEE.
- [34] Lara A, Kolasani A, Ramamurthy B. Network innovation using openflow: A survey. *IEEE communications surveys & tutorials*. 2013 Aug 30;16(1):493-512.
- [35] Bilen T, Canberk B, Chowdhury KR. Handover management in software-defined ultra-dense 5G networks. *IEEE Network*. 2017 Jul 28;31(4):49-55.
- [36] *Network Performance Objectives for IP-Based Services*, Standard ITU-T Recommendation Y.1541, 2011.
- [37] Mali BJ, Ninkovic NM, Stojanovic MD, Savic GI. Service class mapping based on integer programming algorithm in the third party agent. In 2014 22nd Telecommunications Forum Telfor (TELFOR) 2014 Nov 25 (pp. 170-173). IEEE.
- [38] Melo, Francisco S. "Convergence of Q-learning: a simple proof"
- [39] Matiisen, Tambet (December 19, 2015). "Demystifying Deep Reinforcement Learning". *neuro.cs.ut.ee. Computational Neuroscience Lab*. Accessed 2018-04-06.
- [40] Stojanovic MD, Rakas SV. Policies for allocating performance impairment budgets among multiple IP providers. *AEU-International Journal of Electronics and Communications*. 2013 Mar 1;67(3):206-16.
- [41] Mali BJ, Ninkovic NM, Stojanovic MD, Savic GI. Service class mapping based on integer programming algorithm in the third party agent. In 2014 22nd Telecommunications Forum Telfor (TELFOR) 2014 Nov 25 (pp. 170-173). IEEE.
- [42] Ninkovic NM, Mali BJ, Stojanovic MD, Savic GI. Multi-objective third-party approach for service class mapping among multiple providers in the internet. *Elektronika ir Elektrotechnika*. 2015 Apr 9;21(2):80-4.
- [43] Wang G, Zhao Y, Huang J, Wu Y. An effective approach to controller placement in software defined wide area networks. *IEEE Transactions on Network and Service Management*. 2017 Dec 20;15(1):344-55.